

# Interlopers, Translators, Scribes, and Seers: Anthropology, Knowledge Representation and Bayesian Statistics for Predictive Modeling in Multidisciplinary Science and Engineering Projects

Deborah Leishman and Laura McNamara  
Statistical Sciences Group  
Los Alamos National Laboratory D-1, Mew Mexico, USA

## **Abstract**

Multidisciplinary projects often lack integrated representations to support a diverse community's problem-solving process. In this paper, we discuss an interdisciplinary approach to knowledge elicitation, representation and transformation developed in the Statistical Sciences group at the Los Alamos National Laboratory. This approach is called Information Integration Technology (IIT), and it meshes techniques from cultural anthropology, the AI community, and Bayesian statistics to address the complexities of multidisciplinary research. Specifically, we use elicitation techniques derived from cultural anthropology to elicit tacit problem-solving structures from the “natives” – generally, the scientists and engineers collaborating on difficult R&D problems. The elicited information, in turn, is used to develop ontologies that both represent the problem space in the “native language” of the research team, but which are more mathematically tractable to AI and statistical communities. Iterative cycles of representational refinement and quantification lead to the emergence of predictive statistical models that make intuitive sense to all parties: the engineers, elicitation experts, knowledge modelers and statisticians. This method can be used in many types of problems including reliability quantification as shown here.

## **1. INTRODUCTION**

Statisticians are often asked to provide predictive risk and reliability assessments for a wide range of research and development projects. When these projects are very innovative, however, the statistician may be faced with the dilemma of minimal data for the system under scrutiny. Complicating such situations is the increasing ubiquity of multidisciplinary and multinational research teams: statisticians often find themselves asked to contribute to complex, emergent projects that challenge their ability to build predictive models capable of integrating multiple types of data, information and knowledge from a wide range of sources.

In this paper, we discuss an interdisciplinary approach to knowledge elicitation, representation and transformation developed in the Statistical Sciences group at the Los Alamos National Laboratory. This approach is called Information Integration Technology (IIT), and it meshes techniques from cultural anthropology, the AI community, and Bayesian statistics to address the complexities of multidisciplinary research. Specifically, we use elicitation techniques derived from cultural anthropology to elicit tacit problem-solving structures from the “natives” – generally, the scientists and engineers collaborating on difficult R&D problems. The elicited information, in turn, is used to develop ontologies that both represent the problem space in the “native language” of the research team, but which are more mathematically tractable to the AI and statistical communities. Iterative cycles of representational refinement and quantification lead to the emergence of predictive statistical models that make intuitive sense to all parties: the engineers, elicitation experts, knowledge modelers and statisticians.

In the following pages, we describe the origins and structure of the IIT approach and demonstrate its use in the development of a hierarchical reliability model for a complex rocket system. The IIT knowledge modeling techniques are of particular interest to Bayesian statisticians, whose problem solving approach often relies on complex hierarchical networks.

## **2. A PRIMER ON STATISTICAL CONSULTING AND BAYESIAN HIERARCHICAL MODELING**

Statisticians who work in experimental science and engineering fields become quite adept at consulting with research teams to develop a wide range of probabilistic models for decision-making. Traditionally, statisticians have worked fairly bounded pieces of a larger problem: experimental design, for example, or failure mode analysis. This trajectory has resulted in a standard model for statistical consulting in which the clients provide the statistical consultant with a problem definition and some data sources that, in the statistician’s mind, lend themselves to a particular class of models. The statistician goes back “over the fence” and works an area of the problem, periodically asking clients to clarify some aspect of the model or to provide additional data.

The past twenty years or so, however, have seen a trend towards large-scale, complex, multidisciplinary scientific projects that often incorporate experts from a wide range of disciplines, including engineering, biology, physics, computer science, chemistry, and others. The complexity of these problems often requires a greater level of participation from the statistician and also demands a statistical approach capable of combining multiple forms and types of data.

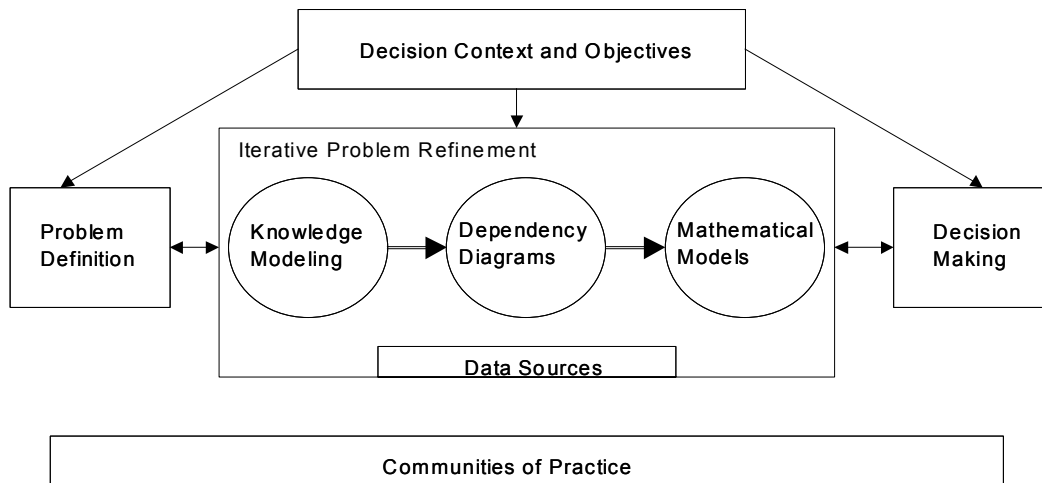
Bayesian statistics is one such approach. This relatively new subfield of statistics was perceived until recently among more traditional “frequentist” statisticians as a radical, controversial, and even untenable approach to estimating probability (Wilson 2001). Today, Bayesian models are widely used to combine multiple sources of data to estimate the probability of an event in the future, based on relevant information regarding the occurrence of that event in the past. Although Bayesian models are well suited to addressing complex problems, constructing a Bayesian model requires a great deal of time and information about the problem at hand. The IIT approach was designed to address this problem by using Conceptual Graphs to represent the complex problem space. Because Bayesian models are

represented as chain graphs (i.e., nodes connected by arcs), they are remarkably synergistic with Conceptual Graphs.

### 3. INFORMATION INTEGRATION TECHNOLOGY FRAMEWORK

The diagram shown below in Figure 1 outlines the IIT framework, which we use to derive qualitative knowledge models of a domain of interest, and transform these knowledge models into quantitative mathematical models such as Bayesian networks. The framework specifies the context in which these models are being formulated: for example, a decision-making environment in which they will be used to predict the reliability or performance of a system.

IIT methods and the IIT framework are designed to support the emergence of a comprehensive, quantitative decision support model through developing a set of knowledge representations that serve as a common denominator for all problem owners. In a complex system reliability problem, “problem owners” may include engineers, program managers and sponsors, computer scientists, physicists, technicians, and other experts contributing to the problem. IIT requires the ongoing involvement of a knowledge modeler, who acts as a translator working iteratively among the problem owners, technical experts and consultant statisticians. The resulting graphical models provide a comprehensive, nuanced representation of the problem space. These representations are arranged hierarchically in interlinked levels of abstraction, the highest of which provides problem owners with an overview of the entire problem space. The hierarchy of specification enables project participants to drill more deeply into important areas of the problem while maintaining a consistent logical structure throughout all levels of problem representation.



**Figure 1. The Information Integration Technology Framework.**

The first stage in the IIT method is elicitation of the foundation elements: identifying the communities of practice and/or stakeholders involved in the problem, defining the problem space and the decisions that are to be made by all stakeholders, and documenting the

relationship between the stakeholders' objectives and their decisions. Once the problem space is defined, the knowledge modeler begins to work with experts to elicit the conceptual structures they use to work the problem. Using this elicited information, the knowledge modeler develops graphical representations of the problem space using those elements. The visual representations used in the IIT method are derived from conceptual graph techniques pioneered by John Sowa (1984). As these qualitative representations emerge, the knowledge modeler works iteratively with the problem owners, experts and consultant statisticians to formulate the dependencies between concepts in the knowledge model. Once finalized with the experts, the knowledge modeler and the statistician begin transforming these qualitative representations into mathematical models.

The resulting mathematical framework is an extremely useful structure capable of combining multiple types of quantitative information to support decisionmaking in a traceable manner. Doing so requires identifying appropriate data sources to populate nodes in the model, transforming these data into joint probability distributions, and propagating these distributions and their associated uncertainties through the model.

## 4. CONCEPTUAL GRAPHS

The conceptual graph model proposed by John Sowa (1984) is a method of representing the mental models that people use to understand the world. This approach combines a mapping to and from natural language with a mapping to logic. A conceptual graph, which consists of concepts and relations connected by arcs, asserts a proposition and takes the form of a finite connected bipartite graph. *Concepts* represent any entity, attribute, action, state or event that can be described in natural language. *Relations* detail the roles that each concept plays, and the *arcs* serve as connectors between the two. These graphs can be written in either a graphical representation or in a linear form to conserve space.

### 4.1. Simple Graphs

This section presents parts of the conceptual graph model that form a central core. This includes concepts, relations and the arcs between them. Central to the model is the ability to map the graphs into first order predicate calculus. An example of a simple graph is:

[Cat: #123] -> (State) -> [Sit] -> (Location) -> [Mat]

Which represents "A cat named 123 is sitting on a mat."

#### 4.1.1. Concepts and Relations

Concepts represent the entities, attributes, actions, states or events found in natural language. In conceptual graph notation, they are shown as square boxes. A concept box has a referent field on the right of the colon. In this way both generic concepts and particular individuals can be referred to. For example, [Person: \*] or [Person] both refer to the generic concept contains two fields separated by a colon and shows a concept type on the left of the colon and of Person, while [Person: #123] or [Person: Sam] refer to particular individuals, one named Sam and one named 123. Every generic concept in the graph terminology is existentially quantified. Generic concepts act like variables in logic, while individuals are

like constants in logic. Relations in the conceptual graph model specify the role a concept plays and define the relationship between concepts. Relations are shown as circles in the graph notation and can have any number of arcs. For example (Past) is a monadic relation with one arc, (Agent) is a dyadic relation with two arcs and (Between) is a triadic relation requiring three arcs.

#### 4.1.2. A Logical Mapping

The conceptual graph model defines the operator  $\phi$  which maps simple conceptual graphs into formulas in the first order predicate calculus. For these simple graphs, the only logical operators which are needed are conjunction and the existential quantifier. For example, the conceptual graph:

[Cat: #123] -> (State) -> [Sit] -> (Location) -> [Mat]

maps into the following formula when the  $\phi$  operator is applied:

$\exists x \exists y (\text{Cat}(\#123) \wedge \text{State}(\#123, x) \wedge \text{Sit}(x) \wedge \text{Location}(x, y) \wedge \text{Mat}(y)).$

Conceptual graphs are usually more concise than logical formulas because arcs on the graphs show the connections more directly than variable symbols.

### 4.2. Compound Graphs

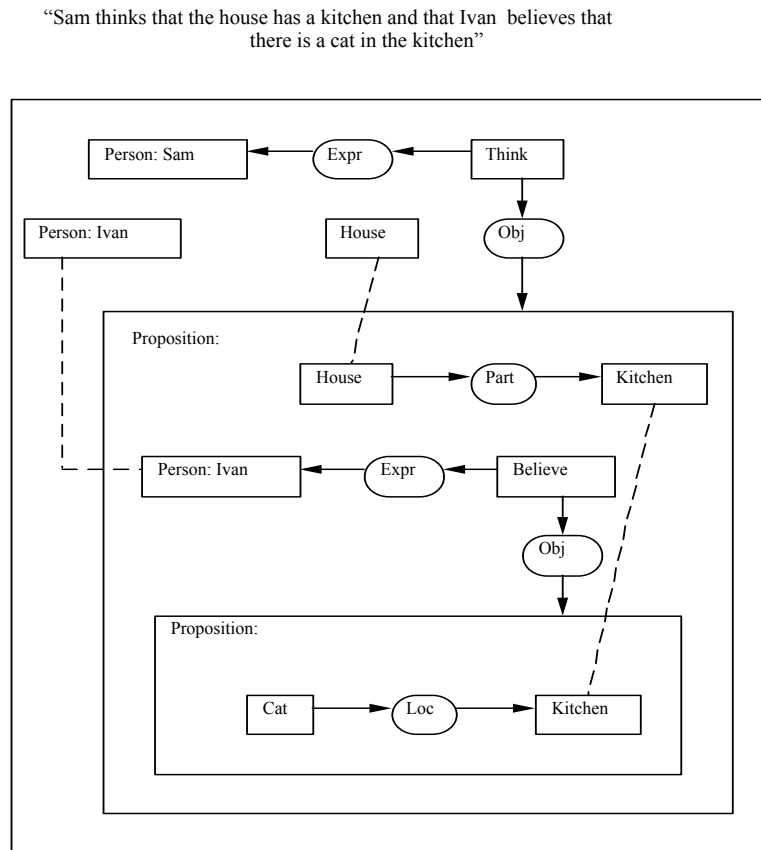
Compound conceptual graphs allow for the expression of more complex sentences than can be described using simple graphs. The components comprising compound graphs are discussed in this section and include nested propositions and co-reference link. Tense, modality and negation can also be represented in Conceptual Graphs. Figure 2 shows an example of a graph that contains most of these elements.

#### 4.2.1. Propositions

A proposition is a concept whose referent is a set of conceptual graphs that are being asserted. The graphs being asserted are said to occur in the context of that proposition, thus propositions are also referred to as context boxes. Propositions can be nested inside of one another and Proposition is the default label for a box that has no other type label. Conjunction of two or more graphs is represented by drawing all the graphs inside a proposition. Figure 2 contains three nested propositions.

#### 4.2.2. Co-reference Links

Co-reference links in conceptual graphs show which concepts refer to the same entities within a graph. In a sentence, these links are expressed as pronouns or other anaphoric references. Figure 2 shows co-reference links using dashed undirected lines. These co-reference links are also referred to by Sowa [1984] as lines of identity and denote an equality relation between concepts. In Figure 2 for example, the phrase, "Sam thinks that the house has a kitchen," refers to a house that Sam already knows about.



**Figure 2. A conceptual graph of a complex sentence.**

## 5. THE CONTEXT: THE ROCKET DEVELOPMENT PROGRAM CENTER

To illustrate the application of the methods we have developed, we use examples from a research and development program that gathers data on test rockets to analyze their performance during flight and to make modifications to their design as necessary. Throughout this discussion, all engineers and agencies are aliased to maintain controls over Proprietary and sensitive information about the program. We refer to this program as the RDP, or the Rocket Development Program.

The oversight agency for the RDP is a group of engineers located in the southeastern United States; we refer to them as RDPC, or the Rocket Development Program Center. Two other groups of engineers are responsible for building separate sections of the rocket: one group of engineers is building a booster to send the rocket into the upper atmosphere, while the other group designs a test payload for the rocket to carry. In addition, several other sub-contractors and vendors provide parts and support to each of the two primary engineering agencies. RDPC is primarily responsible for project management, cost controls, and scheduling.

The RDPC program managers came to Los Alamos with a specific problem: how does one develop a predictive reliability model for an engineering system that is still in the design stages? Multiple concerns drove this question: the rocket development program is extremely

expensive. Only one or two of the prototypes is built and flown and is usually destroyed in the process; rarely are the engineers able to salvage subsystems for reuse in further iterations of the program. Because each system flown is unique, there is little direct, performance, or reliability data available for parts or subsystems on the test rocket. Hence the program managers had little idea how to make predictions or assess risk areas for the flights.

The goal of the LANL/RDP collaboration was to develop an integrated, full-system, predictive reliability model for an upcoming rocket flight. In developing the model, Los Alamos developed a model framework that captured the critical interactions among the rocket's subsystems during flight. We also elicited and document the many sources of data and information that the engineers used to build confidence in their rocket before flight. The resulting model combines multiple sources of information in a rigorous, quantitative framework that can be used to identify and weigh potential risk areas to overall mission "success."

## **6. BUILDING A MODEL**

### **6.1. Engineering Representations**

The contracting engineers in charge of developing the rocket are prolific creators of representations: mechanical drawings, electrical layout diagrams, interface control documents, reliability block diagrams, viewgraphs for debating design issues. Not surprisingly, many of these engineers expressed doubt about the utility of creating even more diagrams of their systems. However, while their representations were sufficient for building a test rocket, they were not sufficient for creating a statistical reliability model. For one thing, engineering drawings – like all representations – are locally meaningful mediums of expression that require experiential knowledge to be sensible to the viewer. Hence it can be quite difficult for a nonmember to decode the representations created by a community of engineers she has only recently met.

The design and development process that the contracting engineers follow compounds this problem. As anthropologist Etienne Wenger (1998) has observed, problem solving is a process of devising representations of knowledge around which parties negotiate meaning. Like many engineering communities, the two primary contractors in the RDPC project each assign bounded teams of engineers to work on separate subsystems of the rocket. Engineering representations are used to communicate design requirements across team boundaries. Each iteration results in new, updated representations that capture the current state of knowledge about each of the subsystems required for a functioning rocket. However, at no point in the engineering problem-solving process does the community develop an integrated representation of the rocket's many subsystems as they are intended to work during flight. Indeed, demonstration of the successful integration of the community's many "ways of knowing" only takes place once the rocket is in flight.

To develop a reliability model as a Bayes net, however, the statistician must understand relationships among different elements of the rocket as it works during flight. This is where knowledge modeling becomes a critical step in creating an integrated model, one that captures subtle dependencies among interrelated parts and uses those dependencies to predict states for the overall mission.

## **6.2. Defining project goals and identifying adviser-experts**

The first step in the IIT knowledge modeling process was to meet with the RDP project leaders to identify specific goals for the rocket system, to get an overview of how the rocket would function, to find out which contractors were responsible for the major areas of the project, and to determine the metrics that the RDP project leaders would use to assess the project's outcomes. At the same time, we devised a general set of goals for the statistical model: to support the rocket project by identifying risk areas, and to provide a quantifiable, traceable statement of risk to upper-level managers in RDPC.

It is impossible to meet project goals without the cooperation of the project's experts, and this requires identifying cooperative insiders who can act as adviser experts to the knowledge modeling team. To ensure the participation of adviser experts throughout the project, RDPC instructed the lead engineer in each contracting organization to support the model building effort. RDPC also provided funding to these agencies so that they could pay their staff to contribute to the model development. These individuals would serve as adviser-experts within the contract organizations: insiders who would willingly partner with the knowledge modeler to identify other experts and to develop sound elicitation protocols and instruments.

## **6.3. Scratchnets and Success and Failure for System Builders**

Once we had met with the adviser experts in each of the contracting agencies and explained the goals of the project, the next step was development of a formal ontology to represent the primary concepts of knowledge in the problem space, and to understand the network of relationships among those concepts. During our first series of meetings with advisor experts from RDP, the booster contractor, and the payload contractors, we elicited information using a scratchnet (Paton et al, 1994). Scratchnets are straightforward, non-hierarchical node-and-arc drawings that simply identify concepts as related to specified domain.

In addition to developing a scratchnet representation of the problem, we also worked with the problem owners to elicit definitions of success and failure for the RDPC program managers. We borrowed a common aerospace terminology for describing mission outcomes: a "stoplight chart," which is perhaps more accurately described as a continuum of failure-to-success, represented by red, yellow and green panels. Equally important in this stage was eliciting how the booster and payload builders defined success and failure, so that we could understand how their goals interlocked with RDPC's goals. We used the same stoplight continuum in elicitation sessions with our adviser experts at each agency. All "stoplight" charts were ultimately combined into a single chart, with all mission goals and states for mission outcomes clearly mapped. In addition, we worked with RDPC to elicit metrics that would determine each of the states for mission success and failure, while eliciting metrics for subsystem performance from the adviser experts at each contracting agency. This information provided the statisticians with a means of quantifying a range of potential outcomes for each of the subsystems in the rocket, and a way to quantify overall mission success and failure.

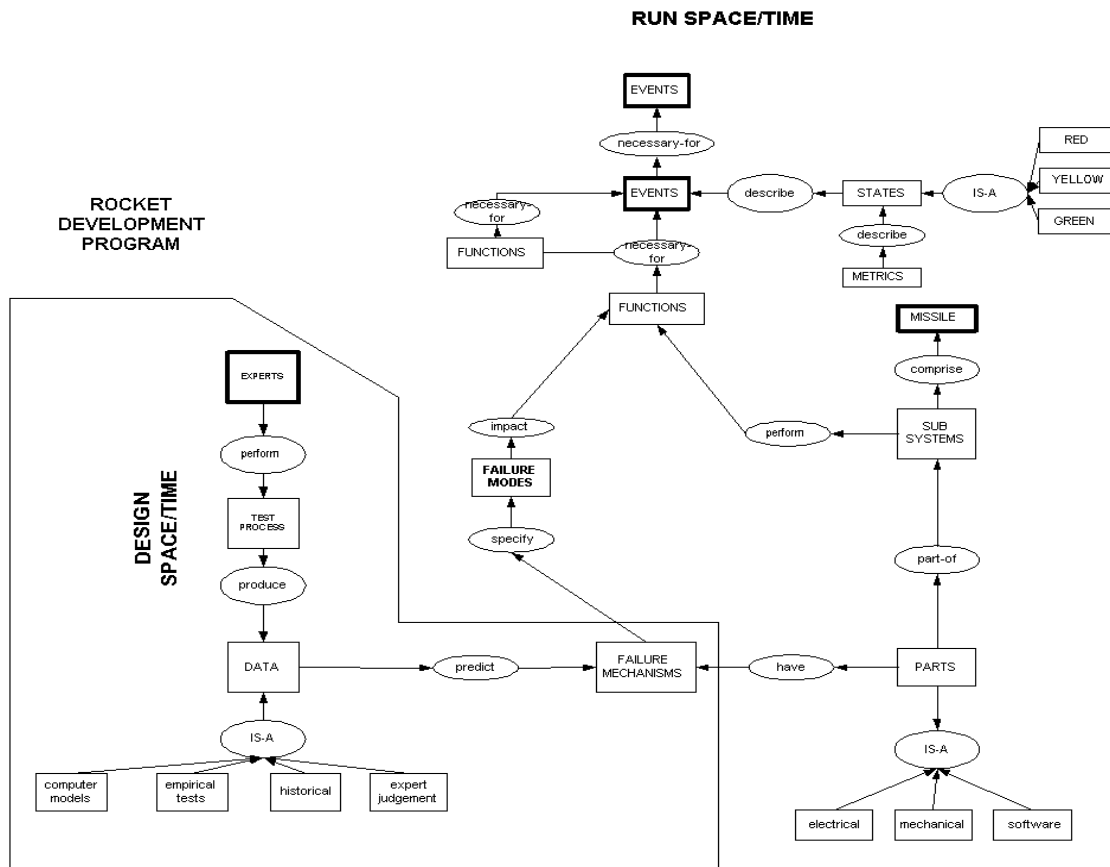
## **6.4 . The Top Level Ontology**

Iteratively refining the scratchnets and the success failure continuums is a learning process for the knowledge modelers and leads to the development of a first-order ontology, one that mapped at the most basic level the key concepts for the domain "RDP-2 Rocket" and the



relationships among those concepts. In the ontology shown in Figure 3, we use a Conceptual Graph representation with concepts as rectangular nodes and relations as circular nodes and arcs indicating directionality among concepts and the relationships that tie them together. The diagram is black and white, but in the actual ontology, the concept boxes were color coded to ensure that specifications of concepts in later drawings were linked to the correct conceptual category. This representation is also recognizable to Bayesians statisticians, who use directed acyclic graphs as structures for propagating uncertainties.

Note that the ontology differentiates between two stages in the design process: “design time,” when the engineers are working to plan and build the rocket; and “run time,” which represents the actual functioning of the rocket during flight. Essentially, the knowledge modeler partnered with the engineers in the “design time” area of the ontology to create a statistical model that would be used to predict the reliability and performance of the rocket system during “run time,” the actual flight. Information generated during the design process in the “design time” area of the ontology was used to create a model structure and to gather data to populate the model.



**Figure 3. Ontology for RDP-2 rocket model.**

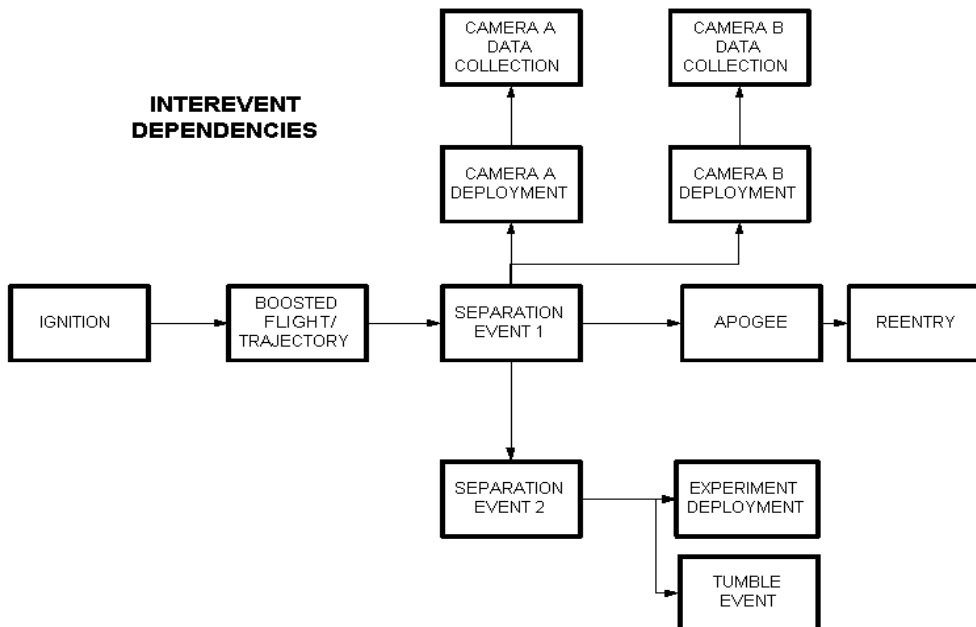
The top-level ontology is a significant point in the IIT method, for it is an elicitation tool that provides a guide for specifying further levels of the domain. In the rocket project, the ontology revealed key focus areas: for example, what functions were required in order for a

particular event to occur? What parts were required for that function to occur? How could failures in individual parts contribute to failed events?

During the elicitation process, the ontology also guides the development of a hierarchy of representations for the problem, from the most general and abstract representation (the top level ontology) to the most specific representations (dependency diagrams that detail specific relationships among parts, subsystems and functions). One critical outcome for the representations is traceability from level to level, so that the representations flow in an orderly fashion from the ontology and make intuitive sense to all parties: the knowledge modelers, the statisticians, RDPC, and the builders of the booster and the payload.

### 6.5. Ontology Specification: Event Dependencies

Once the top level ontology was completed we were ready to begin developing specific representations of its concepts. The first level of specification focused on identifying measurable flight-time events that would act as conceptual waypoints, to discretize the linear flow of the planned rocket trajectory into a series of measurable focus areas for the model. Significantly, the order in this representation of flight events was not a time-ordered linear sequence, but rather a sequence of dependencies as shown in Figure 4. In other words, this level specified the order in which any particular event during the flight could impact, or be impacted, by any other event. Using the success and failure chart in combination with the event specification, the RDP staff could heuristically begin to relate overall mission success to states for any single event, by asking how a red, yellow or green state for a particular event might impact subsequent flight events.

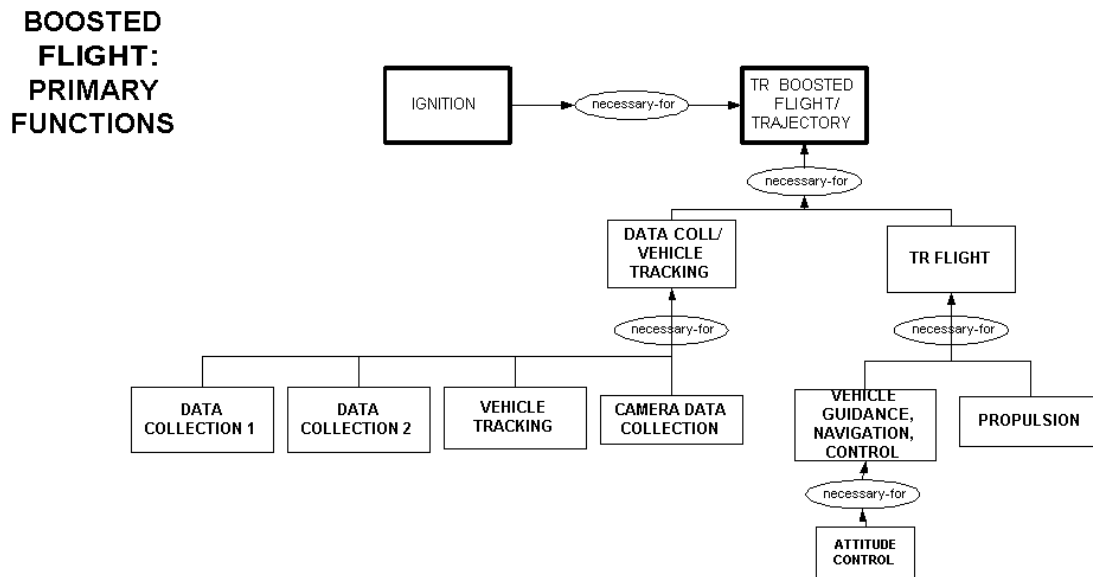


**Figure 4. Specification of inter-event dependencies for rocket flight.**

## 6.6. Functional, Subsystem-Part, and Series-Parallel Specifications

The next stage in specifying the full ontology was to focus on each flight event and begin identifying key parts, subsystems and functions. Working with the subsystem engineers, we created the next three levels of specification for each event: a *functional diagram* that detailed only the functions required for an event; a *subsystem-part diagram* that broke subsystems into collections of parts; and a modified *series parallel diagram* that specified the order in which parts in a subsystem work together to perform a function.

For each event displayed on the Inter-event Dependency Diagram, we created a representation to detail relationships between functions and events. For example, Figure 5 details the functions that the booster must execute in order for the first stage of the flight to occur. Note that the representation says nothing about the *state* (red, yellow, green) of the functions, or the event itself: the functional drawing simply relates functions to other functions and ultimately to the event, “boosted flight.”



**Figure 5. Functional view of event, “Boosted Flight”**

The representation above identifies two primary functions for “TR Flight.” These functions include “Data Collection/Vehicle Tracking,” and “Boosted Flight,” which are themselves broken into several sub-functions. These sub-functions, in turn, can be further specified by the parts and subsystems involved in their performance.

Note that in the drawing above, the event “TR Flight” depends not only on a set of nested functions, but also on a previous event in the trajectory, “Ignition.” Given that a rocket flight is an enormously complex set of dependencies, one of the convenient things about this type of representation is that it allows the knowledge modeler to detail only the functions specifically required for the event in question. In other words, while a “Boosted Flight” of course depends heavily on what happens during “Ignition,” those ignition-related functions are detailed in a set of representations for the “Ignition” event and do not need to be re-drawn for “Boosted Flight.”

Not shown in this paper are the next two levels of representational abstraction. *Subsystem-part representations* are graphical inventories of specific parts and the subsystems that house them. It is important to point out that this view provides no information about *how* any constellation of parts performs a function, but rather identifies how specific parts are grouped into subsystems. This is important since functions are not infrequently the result of individual parts in separate subsystems working simultaneously across subsystems to produce a particular function. This diagram is less a representation for the statistical model than it is a “laundry list” that the knowledge modelers and the engineers use to ensure that all parts are properly grouped into their respective subsystems.

Process knowledge is specified in the next stage of abstraction, a *series parallel diagram* that locates parts within a subsystem and displays the order in which parts function with each other to perform a given function. Most engineering drawings tend to be structural in nature, not functional: in other words, they display connections among parts, rather than describe how parts work together to perform one or more functions. Although we realized that a functional view of the system would be critical for developing any kind of predictive model of rocket performance, that knowledge was not only tacit; it was also distributed across numerous individual engineers. Hence it was necessary to elicit and represent this information using the functional and structural specifications described above. This stage marked the beginning of the transition from an engineering understanding of the system, to a statistical dependency model that could be quantified and populated with available data to make predictions about the rocket in flight.

The series parallel diagram was the first step in this transition. This type of drawing is somewhat similar to a series parallel diagram exemplified in a classic reliability block diagram, but with a great deal more descriptive information. Block diagrams simply connect parts to parts in the order that they must perform so that a given phenomenon occurs. The series parallel diagrams we developed followed the structure of a reliability block diagram but contained a great deal more information about the context of a particular part and its functions.

## **7. DEPENDENCY DIAGRAMS: FROM KNOWLEDGE MODELING TO BAYES NETS**

### **7.1. Dependency Diagrams: Roll Control**

Although different kinds of series parallel diagrams provide a wealth of information about how parts and subsystems and functions are linked to events on the rocket trajectory, these diagrams are not sufficient for building a Bayes net. This is because Bayes nets represent *dependencies* among their elements: given what I know about one node in a model, what might I be able to say about nodes whose states depend on that event? The final stage in the knowledge modeling process, then, is to transform the series parallel diagrams into dependency diagrams. The difference between the two is subtle, but critical: Series parallel diagrams specify the linkages among parts related to a function and imply some order to those parts: for example, a power function might be described as, ‘Battery A feeds power to a PTS, which sends a current to the following electrical components:..’ A dependency diagram, on the other hand, describes that same power function as dependent on the performance of

Battery A *and* the PTS, and how downstream components' performance is (at least partially) dependent on that power function.

The most immediate difference between a basic series parallel diagram and a dependency diagram is that subsystems are not represented in the latter. This is because subsystems simply designate the geographical location of parts within the rocket; dependencies exist between their parts and one or more functions. Strictly speaking, no functions depend on a subsystem; however, many functions may depend on the individual parts *within* a subsystem.

In a dependency diagram, we are concerned with specifying three types of information: how functions depend on one or more necessary parts, how the performance of a particular part depends on a particular function (recursive relationships), and how parts may provide redundancy (part A *or* part B is necessary for function X) or single points of failure (part A *and* part B are necessary for function X). These relations among parts and functions specify the dependency structure for a Bayes net.

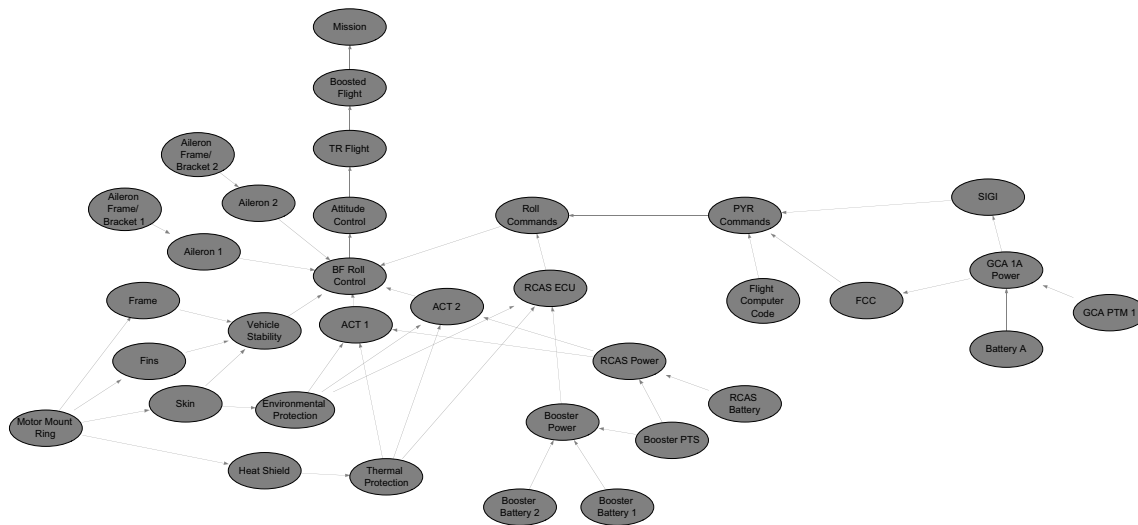
## **7.2. Roll Control: An Example of a Bayes Net**

The final transition occurred when the dependency diagram was turned into the Bayes net structure. The diagram shown in Figure 6 is a Bayes net, extracted from the larger rocket model. The statistician built it using the dependency diagram developed. The initial translation can be performed easily from the dependency diagram to the Bayes net, although the knowledge modeler and the statistician do work together to check the Bayes net and ensure that the statistician has specified the right dependencies, labeled the functions and parts correctly, and indicated the proper directionality in the relationship arcs.

The Bayes net is a highly distilled version of the dependency diagram: it eliminates all relationship labels and, at the level shown above, offers no information about subsystem location for any of the parts. Population of the model occurs in later iterations, using the series parallel diagrams for failure (to designate a range of states for each of the part and function nodes), the stoplight charts (to designate states for the mission events), and the series-parallel data diagrams (to identify sources of data for each part and its associated failure modes). The model generates a probability distribution for each event in the inter-event dependency diagram, as well as a final probability distribution for states red, yellow, and green for the entire mission. In addition, the Bayes net allows the user to trace sample paths for different solutions through the states of each node, so that it is possible to connect given outcome for the entire system to the state of any particular node.

## **8.0 CONCLUSION**

Multidisciplinary projects often lack integrated representations to support the community's problem-solving process. It is frequently difficult for project insiders to develop these representations: for one thing, they are focused on meeting the project's goals. More subtly, insiders often have a great deal of local knowledge about a specific area within a project, but may have difficulty leveraging that into a global view of the problem. Anthropologists and knowledge modelers, on the other hand, are trained to elicit this information and can draw on a wide range of representation techniques to create useful abstractions of the project area.



**Figure 6. Bayes Net Representation for Roll Control**

An interdisciplinary approach to knowledge modeling, one that combines techniques from anthropology, artificial intelligence, and knowledge representation, is particularly helpful in situations where problems are undergoing definition, are emergent, and that involve multiple players from different disciplines and/or geographical locations. When such modeling techniques are paired with quantitative tools from statistics, it becomes possible to develop complex models that can, among other things, enable the integration of multiple, diverse sources of data to estimate performance without testing. Other research-related applications that we are exploring include development of models to quantify the value of an experiment without testing, to estimate the probability that an invader into a secure facility will be interdicted, and to quantify production requirements as a new consumer product is undergoing design and development.

## REFERENCES

Paton, R.C., Lynch, S., Jones, D., Nwana, H.S., Bench-Capon, T.J.M. & Shave, M.J.R. 1994. *Domain Characterisation for Knowledge Based Systems*, Proceedings of A.I. 94 - Fourteenth International Avignon Conference, Volume 1, 41-54.

Sowa, John. 1984. *Conceptual Structures*. Addison Wesley Publishing Co. Reading Ma. 1984.

Wenger, Etienne. 1998. *Communities of Practice: Learning, Meaning and Identity*. Cambridge, UK: Cambridge University Press.

Wilson, Gregory D. *Articulation Theory and Disciplinary Change: Unpacking the Bayesian-Frequentist Paradigm Conflict in Statistical Science*. Dissertation. New Mexico State University, 2001. Ann Arbor: UMI, 2001. AAT3016344.